TAIWANESE JOURNAL OF MATHEMATICS
Vol. 14, No. 3A, pp. 973-998, June 2010
This paper is available online at http://www.tjm.nsysu.edu.tw/

NUMERICAL ALGORITHMS FOR THE LARGEST STRUCTURED SINGULAR VALUE OF A μ–SYNTHESIS CONTROL SYSTEM

Kun-Chu Chen, Chern-Shuh Wang* and Ching-Chang Yen

Abstract. Numerical algorithms for the computation of an upper bound of the largest structured singular value arising from the μ-synthesis control problem are developed. Since the computation for the largest structured singular value has been shown to be an NP-hard problem in literatures, we concentrate the study on the computation for an upper bound of the largest structured singular value. A Newton's type method is proposed. Some theoretical results related to the method are investigated. Numerical implementation shows the efficiency of the method.

1. INTRODUCTION

In this paper, we study the numerical algorithms for the computation of an upper bound of the largest structured singular value corresponding to a rational matrix, say $M(s)$. Consider an internally stable feedback system as given in Fig. 1, where $M(s) = C(s\mathrm{I} - A)^{-1}B \in \mathbb{C}^{m\times m}$ is a transfer function matrix and $\triangle(s)$ is an $m \times m$ matrix function modelling uncertainty. An intuitive question is how large $\triangle(s)$ may go before it causes instability of the closed feedback system in Fig. 1. It means that we are interested in the margin $\| \triangle \|_\infty$ so that the closed loop system has poles in the closed right complex plane.

Since the poles of the closed loop system are determined by the roots of $\det(I - M(s) \triangle (s)) = 0$, for an internally stable $M(s)$, the closed loop system is stable provided that $\| \triangle \|_\infty$ is sufficiently small. The small gain theorem [30] shows that the margin of stability can be determined by $\dfrac{1}{\|M\|_\infty}$ whenever $\triangle(s) \in \mathbb{C}^{m\times m}$ is unstructured. Here $\|M\|_\infty$ is defined by

$$\|M\|_\infty = \sup_{s\in\overline{\mathbb{C}}_+} \overline{\sigma}(M(s)) = \sup_{\omega\in\mathbb{R}} \overline{\sigma}(M(\widehat{i}\,\omega))$$

Received November 30, 2008, accepted January 12, 2009.
2000 *Mathematics Subject Classification*: 65F15, 65F35, 65K15.
Key words and phrases: Largest structured singular value, μ-synthesis, Newton's method, Eigenvalue.
*This work was partially supported by the National Center for Theoretical Sciences in Taiwan.

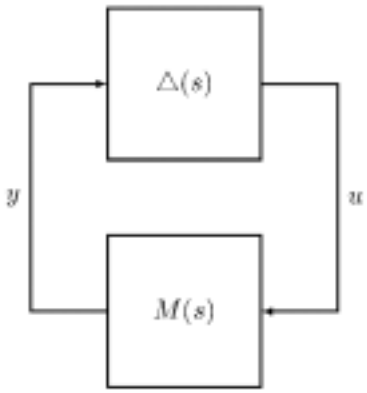

Fig. 1. The block diagram.

which is usually named the $\mathcal{H}_\infty$-norm of $M(s)$, where $\widehat{i} = \sqrt{-1}$. Note that for any fixed $s \in \overline{\mathbb{C}}_+$, the largest singular value of $M(s)$, denoted by $\overline{\sigma}(M(s))$, is equivalent to

$$\overline{\sigma}(M(s)) = \frac{1}{\min\limits_{\triangle \in \mathbb{C}^{m\times m}} \{\overline{\sigma}(\triangle) \mid \det(\mathrm{I} - M(s)\triangle) = 0\}}. \tag{1.1}$$

In other words, $\overline{\sigma}(M(s))$ gives a measure of the smallest *unstructured* uncertainty $\triangle \in \mathbb{C}^{m\times m}$ that causes instability of the closed loop system.

The results of (1.1) can be adopted whenever the uncertainty $\triangle$ is structured. It hence gives the definition of the *largest structured singular value* of $M(s)$ with respect to a structured $\triangle$. Consider the set of the structured uncertainty $\mathbf{\Delta}$ as below.

$$\mathbf{\Delta} = \{\mathrm{diag}(\delta_1 I_{r_1}, \ldots, \delta_S I_{r_S}, \triangle_1, \ldots, \triangle_F) \mid \delta_i \in \mathbb{C}, \triangle_j \in \mathbb{C}^{m_j \times m_j}\}, \tag{1.2}$$

where

$$\sum_{i=1}^{S} r_i + \sum_{j=1}^{F} m_j = m.$$

The largest structured singular value for the matrix $M \in \mathbb{C}^{m\times m}$ with respect to the set of structured uncertainties, $\mathbf{\Delta}$, can be defined as the same as in [30],

$$\mu_{\mathbf{\Delta}}(M) = \begin{cases} \dfrac{1}{\min\limits_{\triangle \in \mathbf{\Delta}} \{\overline{\sigma}(\triangle) \mid \det(I - M\triangle) = 0\}} \\ \\ 0, \text{ if } \det(I - M\triangle) \neq 0 \text{ for all } \triangle \in \mathbf{\Delta}. \end{cases} \tag{1.3}$$

Note that since the triangle inequality is not always true for $\mu_{\mathbf{\Delta}}(\cdot)$, the function $\mu_{\mathbf{\Delta}} : \mathbb{C}^{m\times m} \longrightarrow \mathbb{R}_+ \cup \{0\}$ cannot be a norm. However, the computation of $\mu_{\mathbf{\Delta}}(M)$,

usually plays a key role to the robust stability analysis for a control system having structured uncertainties. Unfortunately, the computation of $\mu_{\mathbf{\Delta}}(M)$ has shown to be an NP-hard problem [5, 28]. Therefore, the study of the computation of upper and lower bounds of $\mu_{\mathbf{\Delta}}(M)$ are ubiquitously blooming in literatures. Some native polynomial-time algorithms for computing upper and lower bounds of $\mu_{\mathbf{\Delta}}(M)$ are assembled already in [2]. Some algorithms for the computation of upper bounds which convert the problems to linear matrix inequalities (LMIs) are available in [10, 16, 18, 26]. On the study of the lower bounds, some literatures are proposed [9, 17, 24, 30]. Using the power iteration, [25, 29] focus on the estimation of the lower bounds. Using these bounds, [18] shows how to estimate the μ-norm for a μ-synthesis system, combining with an elegant algorithm given in [3].

The paper is organized as follows. In Section 2, we summary some preliminaries related to the bounds of $\mu_{\mathbf{\Delta}}(M)$ which have already been proposed in literatures. We also describe the problem to be investigated which is an optimization problem with linear matrix inequality constraints. Section 3 lists a basic algorithm for solving the problem prescribed in Section 2. Some theoretical results related to the development of a Newton's type method are included in Section 4. An appropriate Newton's type algorithm is hence developed in Section 5. The numerical implementation of the developed algorithms is in Section 6, together with some numerical results. Finally, we give some conclusions of this paper in Section 7.

2. Preliminaries and Problem Description

When a particular set of uncertainties $\mathbf{\Delta} = \{\delta I_m \mid \delta \in \mathbb{C}\}$, a set with least degree of freedom, $\mu_{\mathbf{\Delta}}(M) = \rho(M) \equiv \max_{\lambda \in \lambda(M)} |\lambda|$, where $\lambda(M)$ denotes the set of eigenvalues of M. From the definition in (1.1), we have $\mu_{\mathbf{\Delta}}(M)$ satisfying

$$\rho(M) \leq \mu_{\mathbf{\Delta}}(M) \leq \overline{\sigma}(M)\,.$$

To derive more general bounds for $\mu_{\mathbf{\Delta}}(M)$, some sets of uncertainties with higher degree of freedom shall be investigated. Let

$$\mathbb{U} = \{U \in \mathbf{\Delta} \mid UU^* = I_m\},$$
$$\mathbb{D} = \{\mathrm{diag}(D_1, \ldots, D_S, d_1 I_{m_1}, \ldots, d_F I_{m_F}) \mid D_i \in \mathbb{C}^{r_i \times r_i}, D_i = D_i^* > 0, d_j \in \mathbb{R}^+\}.$$

Then, some associated bounds are listed in [30]:

$$\rho(M) \leq \max_{U \in \mathbb{U}} \rho(UM) \leq \mu_{\mathbf{\Delta}}(M) \leq \inf_{D \in \mathbb{D}} \overline{\sigma}(DMD^{-1}) \leq \overline{\sigma}(M).$$

In this paper, we are interested in the computation of the bound $\inf_{D \in \mathbb{D}} \overline{\sigma}(DMD^{-1})$, instead of $\mu_{\mathbf{\Delta}}(M)$ which leads to an NP-hard problem [10]. However, in general,

$\inf_{D\in\mathbb{D}} \overline{\sigma}(DMD^{-1})$ is hard to compute unless a restricted subset of $\mathbb{D}$, denoted by $\mathbb{D}_d$, is adopted, where

$$\mathbb{D}_d = \{\text{diag}(D_1, \ldots, D_S, d_1 I_{m_1}, \ldots, d_F I_{m_F}) \mid$$
$$D_i \in \mathbb{C}^{r_i \times r_i}, D_i = D_i^* \text{ is diagonal and positive definite}, d_j \in \mathbb{R}^+\}$$

Clearly,

$$\mu_{\mathbf{\Delta}}(M) \leq \inf_{D\in\mathbb{D}} \bar{\sigma}(DMD^{-1}) \leq \inf_{D\in\mathbb{D}_d} \bar{\sigma}(DMD^{-1}). \tag{2.1}$$

Straightforward calculations lead to

$$\begin{aligned}\inf_{D\in\mathbb{D}_d} \bar{\sigma}(DMD^{-1}) &= \inf_{D\in\mathbb{D}_d} \{\gamma | (DMD^{-1})^*(DMD^{-1}) \leq \gamma^2 \mathrm{I}\} \\ &= \inf_{D\in\mathbb{D}_d} \{\gamma | M^*D^*DM \leq \gamma^2 D^*D\} \\ &= \inf_{P\in\mathbb{R}_+(\mathbb{D}_d)} \{\gamma | M^*PM \leq \gamma^2 P\}\end{aligned}$$

where $\mathbb{R}_+(\mathbb{D}_d)$ is a subset of $\mathbb{D}_d$ with all diagonal elements being positive real numbers [18].

Define the following matrix functions of vector variable $x = [x_1, \cdots, x_m]^T \in \mathbb{R}_+^m$,

$$A(x) \equiv \sum_{i=1}^{m} x_i A_i, \quad \text{where} \quad A_i = M^* e_i e_i^T M, \tag{2.2}$$

$$B(x) \equiv \sum_{i=1}^{m} x_i B_i, \quad \text{where } B_i = e_i e_i^T \tag{2.3}$$

and

$$e_i \text{ is the } i\text{-th column of the } m\text{-th indentity } I_m,$$
$$\mathbb{R}_+^m = \{x = [x_1, \cdots, x_m]^T \in \mathbb{R}^m, \ x_i > 0\}.$$

Therefore, finding $\inf_{D\in\mathbb{D}_d} \overline{\sigma}(DMD^{-1})$ is equivalent to solving a *linear matrix inequality* (LMI, in short), referred to as the *generalized eigenvalue problem* (GEVP) in [4] as below.

Problem 1.

$$\begin{cases} \min\limits_{x} \lambda_{\max}(A(x), B(x)), & \text{where } \lambda_{\max}(A(x), B(x)) \text{ is the maximal} \\ \text{eigenvalue of } & A(x) - \lambda B(x), \text{ subject to } x = (x_1, \ldots, x_m) \in \mathbb{R}^m_+. \end{cases}$$

Consequently, the upper bound for the largest structured singular value $\mu_{\mathbf{\Delta}}(M)$ in (2.1) can be computed by the fact of

$$\inf_{D \in \mathbb{D}_d} \bar{\sigma}(DMD^{-1}) = [\min_{x \in \mathbb{R}^m_+} \lambda_{max}(A(x), B(x))]^{1/2},$$

whenever Problem 1 is solvable.

3. A Basic Algorithm for Problem 1

There have already been a lot of numerical methods for solving Problem 1; see [4]. We now give a brief description of a basic method, known as the ellipsoid algorithm, which can be considered a higher dimensional bisection method.

The key step of the ellipsoid algorithm is to find the *"cutting plane"* of an associated ellipsoid containing the feasible cone. The cutting plane of a feasible ellipsoid centered at x is usually determined by the norm vector g such that an optimal point of Problem 1 lies in the half-space $\{z \mid g^*(z - x) < 0\}$. The following arguments are mainly from [4].

Without any ambiguity we denote $\lambda_{\max}(x) = \lambda_{\max}(A(x), B(x))$. Pick $u \neq 0$ to be the eigenvector satisfying

$$A(x)u = \lambda_{\max}(x)B(x)u. \tag{3.1}$$

Define $g = [g_1, \cdots, g_m]^T$ by

$$g_i = -u^*(\lambda_{\max}(x)B_i - A_i)u, \ i = 1, \cdots, m. \tag{3.2}$$

From the definitions in (2.2) and (2.3), we know that $A(x)$, $B(x)$ are both Hermitian and $B(x)$ is positive definite, so $\lambda_{\max}(x)$ has to be a real number. Using (2.2), (2.3) and (3.1), we conclude that

$$\begin{aligned} g^*x = \sum_{i=1}^m g_i^* x_i &= -\left[\sum_{i=1}^m u^*(\lambda_{\max}(x)B_i - A_i)ux_i \right] \\ &= -\left[u^*(\lambda_{\max} \sum_{i=1}^m x_i B_i - \sum_{i=1}^m x_i A_i)u \right] \\ &= -u^* \left[\lambda_{\max} B(x)u - A(x)u\right] = 0. \end{aligned}$$

Similar, for any vector z,

$$\begin{aligned} g^*(z-x) = g^*z &= \sum_{i=1}^{m} g_i^* z_i \\ &= -\sum_{i=1}^{m} u^*(\lambda_{\max}(x)B_i - A_i)uz_i \\ &= -u^*(\lambda_{\max}(x)\sum_{i=1}^{m} z_i B_i - \sum_{i=1}^{m} z_i A_i)u \\ &= -u^*[\lambda_{\max}(x)B(z) - A(z)]u. \end{aligned}$$

Hence, if $g^*(z-x) \geqslant 0$, then

$$u^*(\lambda_{\max}(x)B(z) - A(z))u \leqslant 0.$$

This implies

$$\lambda_{\max}(x) = \lambda_{\max}(A(x), B(x)) \leqslant \lambda_{\max}(A(z), B(z)) = \lambda_{\max}(z),$$

whenever $g^*(z-x) \geqslant 0$. Therefore, the vector g in (3.2) gives a normal vector of the cutting plane, and indicates that the optimal point lies in the half-space $\{z \in \mathbb{C}^m \mid g^*(z-x) < 0\}$.

We now give a brief summary of the ellipsoid algorithm, which is also named the bisection algorithm hereafter for solving Problem 1.

Ellipsoid Algorithm (A Higher Dimensional Bisection Algorithm) [4]

Input: M, $x^{(1)} = [x_1^{(1)}, \cdots, x_m^{(1)}]^T \in \mathbb{R}_+^m$, $\mathbb{E}^{(1)} = \text{diag}(2^{2L}, \cdots, 2^{2L})$,
$L =$ a sufficient large number, $\epsilon =$ a small tolerance, $k = 1$.
Output: a minimizer x_* for Problem 1.
Repeat until convergence.
step 1: Evaluate $A(x^{(k)}), B(x^{(k)})$ by using definitions in (2.2) and (2.3).
step 2: Compute the maximal eigenpair $(\lambda_{\max}^{(k)}, u^{(k)})$ such that

$$(\lambda_{\max}^{(k)}B(x^{(k)}) - A(x^{(k)}))u^{(k)} = 0.$$

step 3: Compute $g^{(k)} = [g_1^{(k)}, ..., g_m^{(k)}]^T$ by

$$g_i^{(k)} = -u^{(k)^*}(\lambda_{\max}^{(k)}B_i - A_i)u^{(k)}, \quad \text{for } \ i = 1, 2, ..., m.$$

step 4: Set $\tilde{g} = g^{(k)}/\sqrt{g^{(k)*}\mathbb{E}^{(k)}g^{(k)}}$.
Update

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \frac{1}{m+1}\mathbb{E}^{(k)}\tilde{g}, \\ \mathbb{E}^{(k+1)} &= \frac{m^2}{m^2-1}\left(\mathbb{E}^{(k)} - \frac{2}{m+1}\mathbb{E}^{(k)}\tilde{g}\tilde{g}^*\mathbb{E}^{(k)}\right). \end{aligned}$$

step 5: The stopping criteria:
If $\|x^{(k+1)} - x^{(k)}\| < \epsilon$, accept $x_* = x^{(k+1)}$, stop.
If $e^{2mL-\frac{k}{2m}} < \epsilon$, accept $x_* = x^{(k+1)}$, stop.
step 6: $k = k+1$, return to **Repeat**.

Note that the volume of k-th ellipsoid is always less than $e^{2mL-\frac{k-1}{2m}}$.

Remark. The bisection algorithm is definitely reliable because the method is always convergent. However, a more effective algorithm is still being sought for solving Problem 1. A well-known class of algorithms, the interior point methods, has been developed for solving nonlinear optimization problems [11, 18]. A suitable interior point method for Problem 1 can be found in the MatLab LMI toolbox, with an executable MatLab command "`gevp`" [21]. Since the theory of interior point methods has been well studied, we omit the discussion of this method here. Furthermore, in practice (see Section 6), the efficiency of `gevp` in MatLab is not satisfactory, and other more efficient numerical methods have to be developed.

4. Some Theoretical Results Related to Newton's Method

Let $u(x)$ be the eigenvector corresponding to $\lambda_{\max}(x)$. The Rayleigh-quotient formula shows that

$$\lambda_{\max}(x) = \frac{u^*(x)A(x)u(x)}{u^*(x)B(x)u(x)}. \tag{4.1}$$

Problem 1 can be possibly solved by finding critical points of $\lambda_{\max}(x)$ in the feasible set. That is, it suffices to find a solution of

$$\nabla\lambda_{\max}(x) = 0 \tag{4.2}$$

in the feasible set $\mathbb{R}^m_+ = \{x = [x_1, \cdots, x_m]^T \in \mathbb{R}^m, x_i > 0,\ i = 1, \cdots, m\}$. Instead of Problem 1, we now investigate the Newton-type methods for solving the nonlinear problem (4.2).

Remark. Equation (4.2) yields the critical points of $\lambda_{\max}(x)$, and we need to calculate all the partial derivatives of $\lambda_{\max}(x)$ and $u(x)$ with respect to each component $x_1, \cdots, x_m$. Without loss of the generality, we assume that $\lambda_{\max}(x)$ is a simple eigenvalue. Then $\lambda_{\max}(x)$ gives an analytic function in a neighborhood of x [1]. This guarantees the existence of partial derivatives of $\lambda_{\max}(x)$ and $u(x)$. When $\lambda_{\max}$ is not a simple eigenvalue of $A(x) - \lambda B(x)$, the symmetry of $A(x)$ and $B(x)$ implies that $\lambda_{\max}$ has to be semi-simple. For that case, the existence of those derivatives is an straightforward extension. For more details related to the semi-simple eigenvalue, refer to [1].

The following lemma expresses $\triangledown\lambda_{\max}(x)$ in terms of $\lambda_{\max}$, u and the coefficient matrices A_i, B_i, $i = 1, \cdots, m$.

Lemma 1. *If* $A(x)u(x) = \lambda_{\max}B(x)u(x)$, *and* $u^*(x)B(x)u(x) = 1$, *then for* $i = 1, ..., m$,

$$\frac{\partial\lambda_{\max}(x)}{\partial x_i} = u^*(x)A_iu(x) - \lambda_{\max}u^*(x)B_iu(x). \tag{4.3}$$

Proof. If $u^*(x)B(x)u(x) = 1$, we have $\frac{\partial}{\partial x_i}(u^*(x)B(x)u(x)) = 0$. This implies

$$u^*_{x_i}Bu + u^*B_iu + u^*Bu_{x_i} = 0.$$

Since $(u^*_{x_i}Bu)^* = u^*Bu_{x_i}$, we have

$$2\Re e(u^*_{x_i}Bu) + u^*B_iu = 0, \tag{4.4}$$

Here $\Re e(u^*_{x_i}Bu)$ stands for the real part of $u^*_{x_i}Bu$. From (4.4) and $A(x)u(x) = \lambda_{\max}B(x)u(x)$, we have

$$\Re e(u^*_{x_i}Au) = -\frac{\lambda_{\max}}{2}u^*B_iu.$$

Therefore,

$$\begin{aligned}\frac{\partial\lambda_{\max}(x)}{\partial x_i} &= \frac{\partial}{\partial x_i}(u^*(x)A(x)u(x))\\ &= u^*_{x_i}Au + u^*A_iu + u^*Au_{x_i}\\ &= 2\Re e(u^*_{x_i}Au) + u^*A_iu\\ &= u^*A_iu - \lambda_{\max}u^*B_iu.\end{aligned}$$

■

Lemma 2. *If* $A(x)u(x) = \lambda_{\max}B(x)u(x)$, *and* $u^*(x)B(x)u(x) = 1$, *then for* $i = 1, \cdots, m$, $j = 1, \cdots, m$,

$$\begin{aligned}\frac{\partial^2\lambda_{\max}(x)}{\partial x_j \partial x_i} &= 2\Re e(u^*(x)A_i u_{x_j}(x)) \\ -\frac{\partial\lambda_{\max}(x)}{\partial x_j}u^*(x)B_i u(x) &- 2\lambda_{\max}\Re e(u^*(x)B_i u_{x_j}(x)).\end{aligned} \tag{4.5}$$

Proof. Lemma 1 gives $\frac{\partial\lambda_{\max}(x)}{\partial x_i} = u^*(x)A_i u(x) - \lambda_{\max}u^*(x)B_i u(x)$. This implies

$$\begin{aligned}&\frac{\partial^2\lambda_{\max}(x)}{\partial x_j \partial x_i} = \frac{\partial}{\partial x_j}\left(\frac{\partial\lambda_{\max}(x)}{\partial x_i}\right) \\ &= \frac{\partial}{\partial x_j}\left(u^*(x)A_i u(x) - \lambda_{\max}(x)u^*(x)B_i u(x)\right) \\ &= u^*_{x_j}A_i u + u^*A_i u_{x_j} - \frac{\partial\lambda_{\max}(x)}{\partial x_j}u^*B_i u - \lambda_{\max}u^*_{x_j}B_i u - \lambda_{\max}u^*B_i u_{x_j}.\end{aligned}$$

Since $(u^*_{x_j}A_i u)^* = u^*A_i u_{x_j}$, $(u^*_{x_j}B_i u)^* = u^*B_i u_{x_j}$ and $\lambda_{\max} \in \mathbb{R}$, we have

$$\frac{\partial^2\lambda_{\max}(x)}{\partial x_j \partial x_i} = 2\Re e(u^*A_i u_{x_j}) - \frac{\partial\lambda_{\max}(x)}{\partial x_j}u^*B_i u - 2\lambda_{\max}\Re e(u^*B_i u_{x_j}) \qquad \blacksquare$$

Lemma 3. *Let* $A(x)u(x) = \lambda_{\max}B(x)u(x)$, *and* $u^*(x)B(x)u(x) = 1$. *Then*

$$\sum_{i=1}^{m} u_{x_i}(x)x_i = (-\frac{1}{2} + b\hat{i})u(x) \text{ for some } b \in \mathbb{R} \tag{4.6}$$

where $\hat{i} = \sqrt{-1}$.

Proof. Differentiating both sides of $A(x)u(x) = \lambda_{\max}B(x)u(x)$ yields

$$A_i u + Au_{x_i} = \frac{\partial\lambda_{\max}}{\partial x_i}Bu + \lambda_{\max}B_i u + \lambda_{\max}Bu_{x_i}.$$

This implies

$$(A - \lambda_{\max}B)u_{x_i} = Bu\frac{\partial\lambda_{\max}}{\partial x_i} - A_i u + \lambda_{\max}B_i u.$$

Thus,

$$\begin{aligned}&(A - \lambda_{\max}B)\sum_{i=1}^{m} u_{x_i}x_i \\ &= Bu\left(\sum_{i=1}^{m}\frac{\partial\lambda_{\max}}{\partial x_i}x_i\right) - \left(\sum_{i=1}^{m}A_i x_i\right)u + \lambda_{\max}\left(\sum_{i=1}^{m}B_i x_i\right)u.\end{aligned} \tag{4.7}$$

From Lemma 1, and definitions of $A(x) = \sum_{i=1}^{m} A_i x_i$, $B(x) = \sum_{i=1}^{m} B_i x_i$, we have

$$\sum_{i=1}^{m} \frac{\partial \lambda_{\max}}{\partial x_i} x_i = 0, \tag{4.8}$$

and hence (4.7) becomes

$$(A - \lambda_{\max} B) \sum_{i=1}^{m} u_{x_i} x_i = 0. \tag{4.9}$$

Since $\lambda_{\max}$ is simple, (4.9) implies that $\sum_{i=1}^{m} u_{x_i} x_i$ is parallel to u. We hence conclude that

$$\sum_{i=1}^{m} u_{x_i}(x) x_i = cu(x), \quad \text{for some } c \in \mathbb{C}. \tag{4.10}$$

Differentiating both sides of $u^*(x)B(x)u(x) = 1$ and using the similar arguments of the proof of Lemma 1, we derive

$$\Re e[u^*(x)B(x)u_{x_i}(x)] = -\frac{1}{2} u^*(x) B_i u(x).$$

This implies

$$\begin{aligned} &\Re e \left[u^*(x)B(x)(\sum_{i=1}^{m} u_{x_i}(x)x_i) \right] \\ &= -\frac{1}{2} u^*(x) \left(\sum_{i=1}^{m} x_i B_i \right) u(x) = -\frac{1}{2} u^*(x) B(x) u(x) = -\frac{1}{2}. \end{aligned} \tag{4.11}$$

Hence $\Re e(c) = -\frac{1}{2}$ and (4.10) concludes

$$\sum_{i=1}^{m} u_{x_i}(x) x_i = (-\frac{1}{2} + b\hat{i}) u(x) \quad \text{for some } b \in \mathbb{R}. \qquad \blacksquare$$

Theorem 2. *Let $\{x^{(k+1)}\}_{k=0}$ be a sequence of the Newton's iteration by*

$$x^{(k+1)} = x^{(k)} - [D^2 \lambda_{\max}(x^{(k)})]^{-1} \nabla \lambda_{\max}(x^{(k)}),$$

where $D^2 \lambda_{\max}(x^{(k)}) = \left[\frac{\partial^2 \lambda_{\max}(x^{(k)})}{\partial x_j \partial x_i} \right]_{m \times m}$. Then

$$x^{(k+1)} = 2x^{(k)}, \quad \text{for } k = 1, 2, \cdots.$$

Proof. Using Lemma 2, the i-th component of $D^2\lambda_{\max}(x^{(k)})x^{(k)}$ becomes

$$\begin{aligned}
&\frac{\partial^2\lambda_{\max}}{\partial x_i\partial x_1}x_1+\cdots+\frac{\partial^2\lambda_{\max}}{\partial x_i\partial x_m}x_m\\
&=\sum_{j=1}^{m}\big[2\Re e(u^*A_iu_{x_j})-\frac{\partial\lambda_{\max}(x)}{\partial x_j}u^*B_iu-2\Re e(\lambda_{\max}u^*B_iu_{x_j})\big]x_j\\
&=2\Re e\Big(u^*A_i\Big(\sum_{j=1}^{m}u_{x_j}x_j\Big)\Big)-\Big(\sum_{j=1}^{m}\frac{\partial\lambda_{\max}(x)}{\partial x_j}x_j\Big)u^*B_iu\\
&\quad-2\Re e\Big(u^*\lambda_{\max}B_i\Big(\sum_{j=1}^{m}u_{x_j}x_j\Big)\Big).
\end{aligned}$$

By (4.8), we have $\sum_{j=1}^{m}\dfrac{\partial\lambda_{\max}}{\partial x_j}x_j=0$, hence

$$\sum_{j=1}^{m}\frac{\partial^2\lambda_{\max}}{\partial x_j\partial x_i}x_j=2\Re e\Big[u^*(A_i-\lambda_{\max}B_i)\sum_{j=1}^{m}u_{x_j}x_j\Big]. \tag{4.12}$$

By applying Lemma 3 to (4.12), we have

$$\begin{aligned}
&\sum_{j=1}^{m}\frac{\partial^2\lambda_{\max}}{\partial x_j\partial x_i}x_j\\
&=2\Re e\Big(u^*(A_i-\lambda_{\max}B_i)(-\frac{1}{2}+b\hat{i})u\Big)\\
&=2\Re e\Big(u^*(A_i-\lambda_{\max}B_i)(-\frac{1}{2}u)\Big)+2\Re e\big(u^*(A_i-\lambda_{\max}B_i)ub\hat{i}\big).
\end{aligned} \tag{4.13}$$

Since $\lambda_{\max}\in\mathbb{R}$ and A_i, B_i are Hermitian, $u^*(A_i-\lambda_{\max}B_i)u\in\mathbb{R}$. Thus $(u^*(A_i-\lambda_{\max}B_i)ub\hat{i})$ has to be a pure imaginary number. Using Lemma 1 again, (4.13) becomes

$$\sum_{j=1}^{m}\frac{\partial^2\lambda_{\max}}{\partial x_j\partial x_i}x_j=2u^*(A_i-\lambda_{\max}B_i)(-\frac{1}{2}u)=-\frac{\partial\lambda_{\max}}{\partial x_i}.$$

Thus we can conclude that

$$D^2\lambda_{\max}(x^{(k)})x^{(k)}=-\nabla\lambda_{\max}(x^{(k)}),$$

or equivalently

$$[D^2\lambda_{\max}(x^{(k)})]^{-1}\nabla\lambda_{\max}(x^{(k)})=-x^{(k)},$$

provided that $[D^2\lambda_{\max}(x^{(k)})]^{-1}$ exists. Therefore,

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - [D^2\lambda_{\max}(x^{(k)})]^{-1}\nabla\lambda_{\max}(x^{(k)}) \\ &= 2x^{(k)}. \end{aligned}$$

■

Remark. Since $A(x) = \sum_{i=1}^{m} A_i x_i$ and $B(x) = \sum_{i=1}^{m} B_i x_i$ are both linear of x, $A(\alpha x) = \alpha A(x)$ and $B(\alpha x) = \alpha B(x)$ for any nonzero scalar $\alpha \neq 0$. This implies

$$\det(A(\alpha x) - \lambda B(\alpha x)) = \alpha^m \det(A(x) - \lambda B(x)).$$

Thus

$$\lambda_{\max}(\alpha x) = \lambda_{\max}(A(\alpha x), B(\alpha x)) = \lambda_{\max}(A(x), B(x)) = \lambda_{\max}(x),$$

for $\alpha \neq 0$. This means that $\lambda_{\max}(x)$ is a degree-1 homogeneous function of x. Using the homogeneity of $\lambda_{\max}(x)$, Theorem 2 shows that

$$\lambda_{\max}(x^{(k+1)}) = \lambda_{\max}(2x^{(k)}) = \lambda_{\max}(x^{(k)}),$$

where $\{x^{(k)}\}_{k=1}^{\infty}$ is generated by the Newton's iteration. Hence, the Newton's method cannot be applied to solve the problem $\nabla\lambda_{\max}(x) = 0$ unless the constraint of homogeneity of $\lambda_{\max}(x)$ is considered. Therefore, from now on, we study on the nonlinear problem having homogeneity constraint as below.

Problem 2.

$$\begin{cases} \nabla\lambda_{\max}(x) = 0, \\ \text{subject to } x = (x_1, \cdots, x_m)^T \in \mathbb{R}_+^m \quad \text{and} \quad x_m = 1. \end{cases}$$

5. Newton-type Algorithm for Problem 2

To develop a Newton-type method for solving Problem 2, we need to know the second derivatives, $\frac{\partial^2\lambda_{\max}(x)}{\partial x_j \partial x_i}$, $1 \leq i, j \leq m$, in terms of $\lambda_{max}(x)$, $u(x)$, A_i, and B_i, $(1 \leq i \leq m)$. As a result of Lemma 2, we firstly express $u_{x_i}(x)$ for $i = 1, \cdots, m$.

For the case of $M \in \mathbb{C}^{m\times m}$ we have $A(x)$ and $B(x) \in \mathbb{C}^{m\times m}$. Without loss of the generality, we assume that $\lambda_{\max}(x)$ is a simple eigenvalue of $A(x) - \lambda B(x)$ and $u(x)$ is its associated eigenvector satisfying $u^*(x)B(x)u(x) = 1$.

Differentiating both sides of $A(x)u(x) = \lambda_{\max}B(x)u(x)$ yields

$$A_i u + Au_{x_i} = \frac{\partial \lambda_{\max}}{\partial x_i} Bu + \lambda_{\max} B_i u + \lambda_{\max} Bu_{x_i}.$$

This implies

$$(A - \lambda_{\max}B)u_{x_i} = Bu\frac{\partial \lambda_{\max}}{\partial x_i} - A_i u + \lambda_{\max} B_i u. \tag{5.1}$$

Since $u^*(x)B(x)u(x) = 1$, we have $\frac{\partial}{\partial x_i}(u^*(x)B(x)u(x)) = 0$. This derives

$$u^*_{x_i} Bu + u^* B_i u + u^* Bu_{x_i} = 0,$$

and thus

$$\Re e(u^* Bu_{x_i}) = -\frac{1}{2} u^* B_i u. \tag{5.2}$$

Define $\Phi = \begin{bmatrix} A - \lambda_{\max}B & u \end{bmatrix}$, then (5.1) implies

$$\Phi \begin{bmatrix} u_{x_i} \\ 1 \end{bmatrix} = Bu\frac{\partial \lambda_{\max}}{\partial x_i} - A_i u + \lambda_{\max} B_i u + u. \tag{5.3}$$

Since $\lambda_{\max}$ is assumed to be a simple eigenvalue of the Hermitian matrix pencil $A(x) - \lambda B(x)$, Φ must have full row rank. This implies that system (5.3) is solvable. A particular solution of (5.3) using the pseudo-inverse $\Phi^+ = \Phi^*(\Phi\Phi^*)^{-1}$ is as follows

$$\begin{bmatrix} \hat{u}_{x_i} \\ 1 \end{bmatrix} = \Phi^*(\Phi\Phi^*)^{-1}(Bu\frac{\partial \lambda_{\max}}{\partial x_i} - A_i u + \lambda_{\max} B_i u + u). \tag{5.4}$$

So a general solution satisfying (5.3) can be given as

$$u_{x_i} = \hat{u}_{x_i} + \beta u, \text{ for an arbitrary scalar } \beta. \tag{5.5}$$

It is remarkable the way u_{x_i} appears in $\frac{\partial^2 \lambda_{\max}}{\partial x_j \partial x_i} = 2\Re e(u^* A_j u_{x_i}) - \frac{\partial \lambda_{\max}}{\partial x_i} u^* B_j u - 2\lambda_{\max}\Re e(u^* B_j u_{x_i})$ from (4.5) in Lemma 2. This implies that only the real part of the scalar β in (5.5) has a contribution to $\frac{\partial^2 \lambda_{\max}}{\partial x_j \partial x_i}$. We hence study the case with $\beta \in \mathbb{R}$ only.

Since $\lambda_{\max}$ is simple, it remains to compute $\beta \in \mathbb{R}$ for identifying u_{x_i}. Substituting (5.5) into (5.2) and using $u^*(x)B(x)u(x) = 1$, yields $\Re e(u^* B\hat{u}_{x_i}) + \beta = -\frac{1}{2} u^* B_i u$. This implies

$$\beta = -\Re e(u^* B\hat{u}_{x_i}) - \frac{1}{2} u^* B_i u. \tag{5.6}$$

Consequently, we briefly describe the Newton's method for solving Problem 2.

Algorithm (Newton's Method)

Input: $M \in \mathbb{C}^{m\times m}$, $x^{(0)}=[\ x_1^{(0)}, \cdots, x_m^{(0)}\ ]^T \in \mathbb{R}_+^m$, with $x_m^{(0)}=1$, $\epsilon =$ a small tolerance, $k=1$.

Output: x_* gives the solution of Problem 2. Hence, $x_* \in \mathbb{R}_+^m$ is the minimizer for Problem 1 or (GEVP).

Repeat until convergence.

step 1: Evaluate $A(x^{(k)}), B(x^{(k)})$ by using (2.2), (2.3).

step 2: Compute $\lambda_{max}^{(k)}, u^{(k)}$ such that

$$(\lambda_{\max}^{(k)} B(x^{(k)}) - A(x^{(k)}))u^{(k)} = 0.$$

step 3: Compute $\widehat{\nabla\lambda_{\max}}(x^{(k)}) = \nabla\lambda_{\max}(x^{(k)})(1:m-1)$ by applying Lemma 1.

step 4: Compute u_{x_i} for $i=1,\cdots,m-1$, by solving (5.3) and applying $(5.5), (5.6)$.

step 5: Compute $\frac{\partial^2\lambda_{\max}}{\partial x_j \partial x_i}(x^{(k)}), 1 \le i,j \le m-1$, by applying Lemma 2.

step 6: Let $\hat{x}^{(k)} = x^{(k)}(1:m-1)$. Update

$$\begin{aligned} \hat{x}^{(k+1)} &= \hat{x}^{(k)} - [D_{m-1}^2\lambda_{\max}(x^{(k)})]^{-1}\widehat{\nabla\lambda_{\max}}(x^{(k)}), \\ x^{(k+1)} &= \begin{bmatrix} \hat{x}^{(k+1)} \\ 1 \end{bmatrix}, \end{aligned}$$

where $D_{m-1}^2\lambda_{\max}(x^{(k)})$ is the $(m-1)$-st principal submatrix of $D^2\lambda_{\max}(x^{(k)})$.

step 7: Stopping criteria:
If $\|\hat{x}^{(k+1)} - \hat{x}^{(k)}\| < \epsilon$ or $\|\widehat{\nabla\lambda_{\max}}(x^{(k)})\| < \epsilon$, accept $x_* = x^{(k+1)}$, stop.

step 8: $k = k+1$, go to **Repeat**.

Remark. Formula (5.4) gives a theoretical formula of a particular solution to the equation (5.3). However, in view of the computation, there is no cheap way to accomplish formula (5.4) accurately. Fortunately, there are several reliable and efficient numerical methods for solving (5.3), for instance, the SVD method or QR factorization method [15].

For the case of $M \in \mathbb{R}^{m\times m}$: we have $A(x), B(x) \in \mathbb{R}^{m\times m}$. Let $(\lambda_{\max}, u(x))$ be the eigenpair of the pencil $A(x)-\lambda B(x)$ corresponding to the maximal eigenvalue $\lambda_{\max}$. Assume that $\lambda_{\max}$ is simple and $u^T(x)B(x)u(x) = 1$. Since $M \in \mathbb{R}^{m\times m}$, clearly $u(x) \in \mathbb{R}^{m\times 1}$.

We now express $u_{x_i}(x)$ whenever $(\lambda_{\max}, u(x))$ is computed. From

$$A(x)u(x) = \lambda_{\max}B(x)u(x) \text{ and } u^T(x)B(x)u(x) = 1,$$

differentiating both equations with respect to x_i, yields

$$A_i u + Au_{x_i} = \frac{\partial \lambda_{\max}}{\partial x_i} Bu + \lambda_{\max} B_i u + \lambda_{\max} Bu_{x_i}, \tag{5.7}$$

and

$$2u^T Bu_{x_i} + u^T B_i u = 0. \tag{5.8}$$

Reordering (5.7) derives

$$(A - \lambda_{\max}B)u_{x_i} - Bu\frac{\partial \lambda_{\max}}{\partial x_i} = -A_i u + \lambda_{\max} B_i u. \tag{5.9}$$

Grouping (5.8) and (5.9) in a matrix-vector form, we have

$$\begin{bmatrix} A - \lambda_{\max}B & -Bu \\ u^T B & 0 \end{bmatrix} \begin{bmatrix} u_{x_i} \\ \dfrac{\partial \lambda_{\max}}{\partial x_i} \end{bmatrix} = \begin{bmatrix} -A_i u + \lambda_{\max} B_i u \\ -\dfrac{1}{2} u^T B_i u \end{bmatrix}. \tag{5.10}$$

To verify the system (5.10) is solvable so that u_{x_i} and $\frac{\partial \lambda_{\max}}{\partial x_i}$ are computable, it suffices to show that the matrix $\begin{bmatrix} A - \lambda_{\max}B & -Bu \\ u^T B & 0 \end{bmatrix}$ is nonsingular. Suppose that there is a nonzero vector $\begin{bmatrix} z^T & \gamma \end{bmatrix}^T$ such that

$$\begin{bmatrix} A - \lambda_{\max}B & -Bu \\ u^T B & 0 \end{bmatrix} \begin{bmatrix} z \\ \gamma \end{bmatrix} = 0.$$

Than

$$(A - \lambda_{\max}B)z - \gamma Bu = 0, \quad \text{and} \quad u^T Bz = 0.$$

If $\gamma \neq 0$, then

$$Az = \lambda_{\max} Bz + \gamma Bu.$$

This implies $A(x) - \lambda B(x)$ has a Jordan block corresponding to $\lambda_{\max}$. This is a contradiction, because both $A(x)$ and $B(x)$ are symmetric and $B(x)$ is positive definite.

If $\gamma = 0$, then $\lambda_{\max}(x)$ must be a multiple eigenvalue of $A(x) - \lambda B(x)$. This contradicts to the assumption that λ_{max} is a simple eigenvalue of $A(x) - \lambda B(x)$. However, the assumption is always true in the generic sense. Therefore, for the case of $M \in \mathbb{R}^{m\times m}$, system (5.10) is solvable and u_{x_i}, $\frac{\partial \lambda_{\max}}{\partial x_i}$ are hence computable.

We now give the real-valued Newton's method for solving Problem 2.

Algorithm (Newton's Method for Real-Valued Case)

Input: $M \in \mathbb{R}^{m\times m}$, $x^{(0)}=[\ x_1^{(0)}, \cdots, x_m^{(0)}\ ]^T \in \mathbb{R}_+^m$ with $x_m^{(0)}=1$, $\epsilon =$ a small tolerance, $k=1$.

Output: x_* gives the solution of Problem 2. Hence, $x_* \in \mathbb{R}_+^m$ solves Problem 1, i.e., (GEVP).

Repeat until convergence.

step 1: Compute $A(x^{(k)}), B(x^{(k)})$, by using (2.2), (2.3).

step 2: Compute $\lambda_{\max}^{(k)}, u^{(k)}$ such that

$$(\lambda_{\max}^{(k)} B(x^{(k)}) - A(x^{(k)}))u^{(k)} = 0.$$

step 3: Compute $\widehat{\nabla\lambda_{\max}}(x^{(k)}) = \nabla\lambda_{\max}(x^{(k)})(1:m-1)$ by applying Lemma 1.

step 4: For $\ell = 1, \cdots, m-1$,

Solve $\begin{bmatrix} A-\lambda_{\max}B & -Bu \\ u^TB & 0 \end{bmatrix} \begin{bmatrix} u_{x_\ell} \\ \frac{\partial\lambda_{\max}}{\partial x_\ell} \end{bmatrix} = \begin{bmatrix} -A_\ell u + \lambda_{\max}B_\ell u \\ -\frac{1}{2}u^TB_\ell u \end{bmatrix}$

End for

step 5: Compute $\frac{\partial^2\lambda_{\max}}{\partial x_j \partial x_i}(x^{(k)}), 1 \le i,j \le m-1$, by applying Lemma 2.

step 6: Let $\hat{x}^{(k)} = x^{(k)}(1:m-1)$. Update

$$\begin{aligned} \hat{x}^{(k+1)} &= \hat{x}^{(k)} - [D_{m-1}^2\lambda_{\max}(x^{(k)})]^{-1}\widehat{\nabla\lambda_{\max}}(x^{(k)}), \\ x^{(k+1)} &= \begin{bmatrix} \hat{x}^{(k+1)} \\ 1 \end{bmatrix}, \end{aligned}$$

where $D_{m-1}^2\lambda_{\max}(x^{(k)})$ is the $(m-1)$-st principal submatrix of $D^2\lambda_{\max}(x^{(k)})$.

step 7: Stopping criteria:

If $\|\hat{x}^{(k+1)} - \hat{x}^{(k)}\| < \epsilon$ or $\|\widehat{\nabla\lambda_{\max}}(x^{(k)})\| < \epsilon$, accept $x_* = x^{(k+1)}$, stop.

step 8: $k = k+1$, go to **Repeat**.

Remark. Step 4 of the algorithm which solves an $m+1$ linear system constitutes the main cost of computation for each iteration of Newton's method (for real-valued cases). To save the computation cost, we usually adopt the LU-decomposition of $\begin{bmatrix} A-\lambda_{\max}B & -Bu \\ u^TB & 0 \end{bmatrix}$ for solving the corresponding linear system. The cost is hence about $3m^3$ operations [7].

6. Numerical Implementation

The numerical implementation is accomplished by MatLab codes on a Intel(R) Pentium 4, CPU 2.00GHz machine with 256 MB of RAM.

Example 1. Let

$$M = \begin{bmatrix} 0 & 1 \\ c & b \end{bmatrix}.$$

For this case, whenever a normalized condition is imposed by $x_2 = 1$, the exact solution of Problem 1 is

$$x_* = \begin{bmatrix} c \\ 1 \end{bmatrix}.$$

Let $c = 2$ and $b = 3$. Fig. 2, shows the curve of $\lambda_{\max}$ versus x_1. The perfect convexity of the curve shows that $\lambda_{\max}$ has the minimum at $x_1 = 2$.

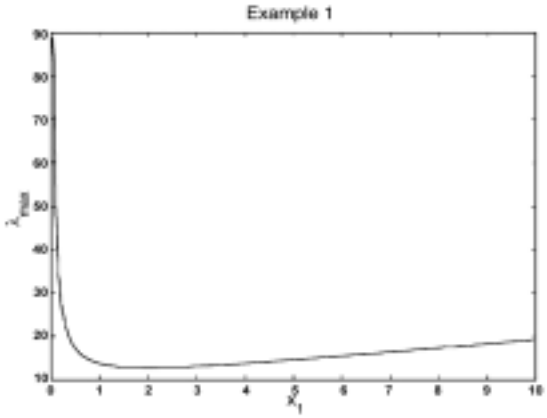

Fig. 2. $\lambda_{\max}$ versus x_1.

Figs. 3, 4 illustrate the numerical results for Example 1 by applying the bisection and Newton's methods, respectively, with the initial guess $\begin{bmatrix} 1 & 1 \end{bmatrix}^T$. The quadratic convergence of Newton's method is realistic in Fig. 4. Fig. 5 gives the comparison of efficiency of the bisection method, Newton's method and the method associated with "gevp" in MatLab LMI Toolbox [21]. Numerical results here indicate that the Newton's is the most efficient one among proposed three methods. The stopping criteria in the numerical experiment for Example 1 are $\|\widehat{x}^{(k+1)} - \widehat{x}^{(k)}\| < 10^{-8}$ and $\|\widehat{\nabla\lambda_{\max}}(x^{(k)})\| < 10^{-8}$, respectively. Note that in all figures of this section, we denote x_* to be the exact solution and x_k to be the k-th iterative vector generated by the corresponding method.

Example 2. The example is from [19]. Consider

$$M = \begin{bmatrix} 4+i & 4i \\ -1 & i \end{bmatrix}.$$

The exact solution is

$$x_* = \begin{bmatrix} 0.25 \\ 1.00 \end{bmatrix}.$$

Figs. 6, 7 illustrate the numerical results for Example 2 by applying the bisection and Newton's method, respectively, with initial guess $\begin{bmatrix} 1 & 1 \end{bmatrix}^T$. The comparison of numerical results of proposed algorithms are illustrated in Fig. 8. The stopping criteria for test numerical methods are $\| \hat{x}^{(k+1)} - \hat{x}^{(k)} \| < 10^{-8}$ and $\| \widehat{\nabla \lambda_{\max}}(x^{(k)}) \| < 10^{-8}$. Once again, Fig. 7 highlights the quadratic convergence of Newton's method. Fig. 8 shows that Newton's method is most efficient among the proposed three methods.

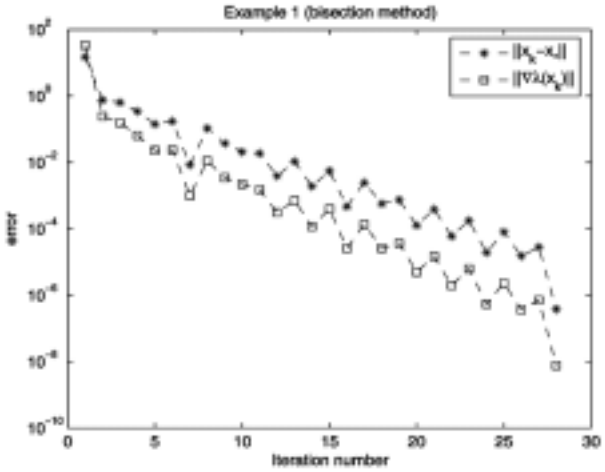

Fig. 3. Numerical results for Example 1 by using the bisection method.

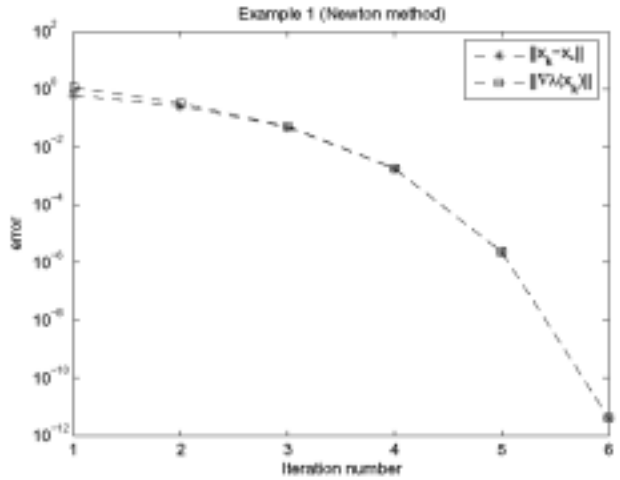

Fig. 4. Numerical results for Example 1 by using Newton's method.

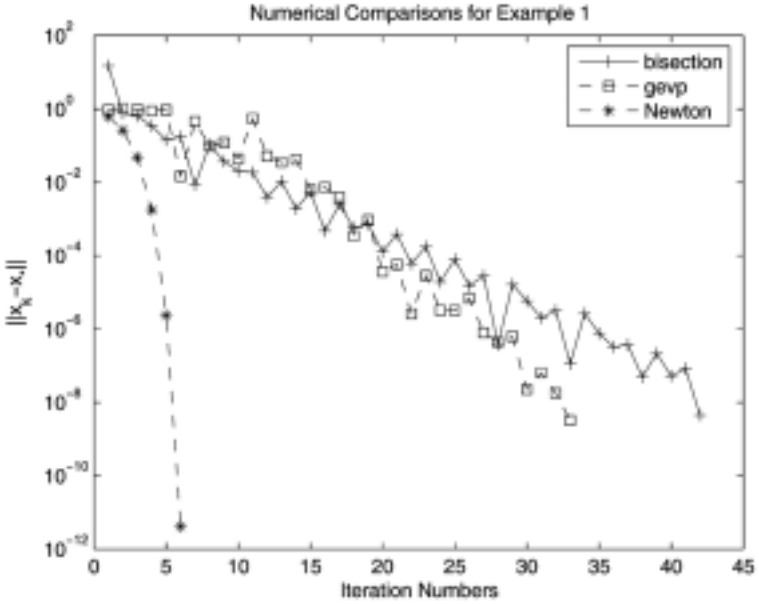

Fig. 5. Comparisons of the efficiency of numerical methods.

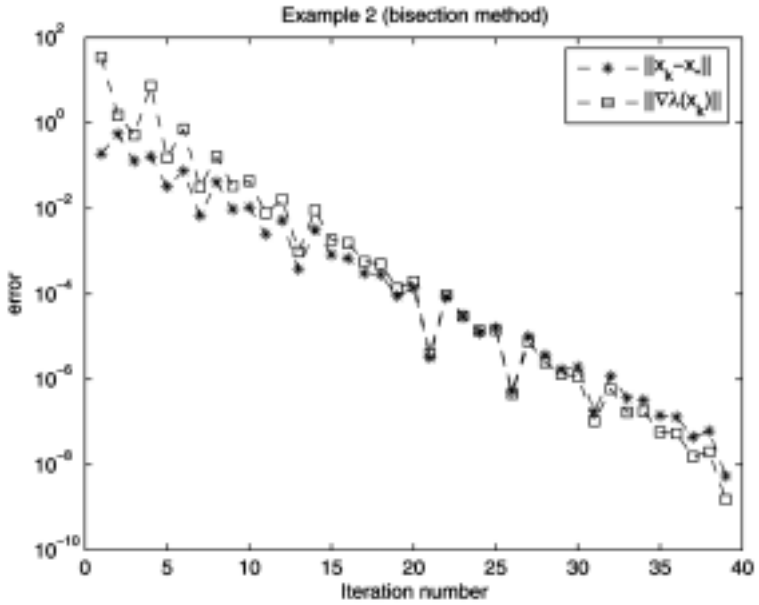

Fig. 6. Numerical results for Example 2 by using the bisection method.

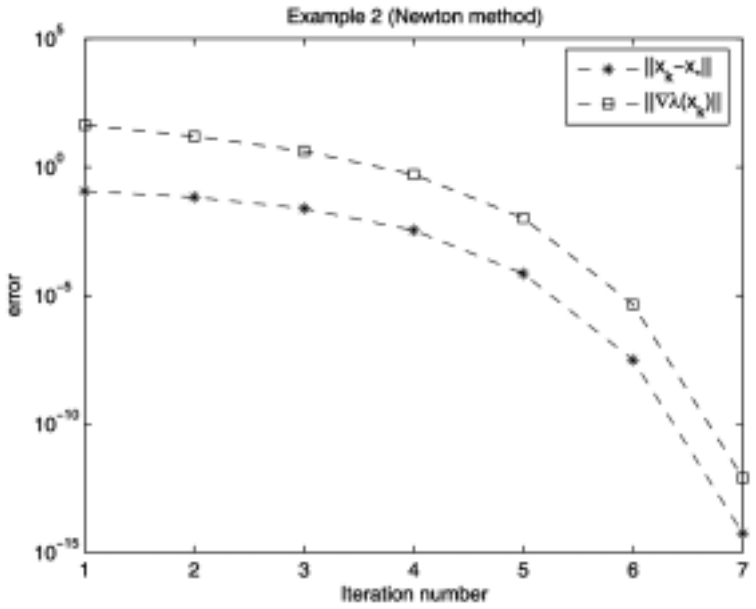

Fig. 7. Numerical results for Example 2 by using Newton's method.

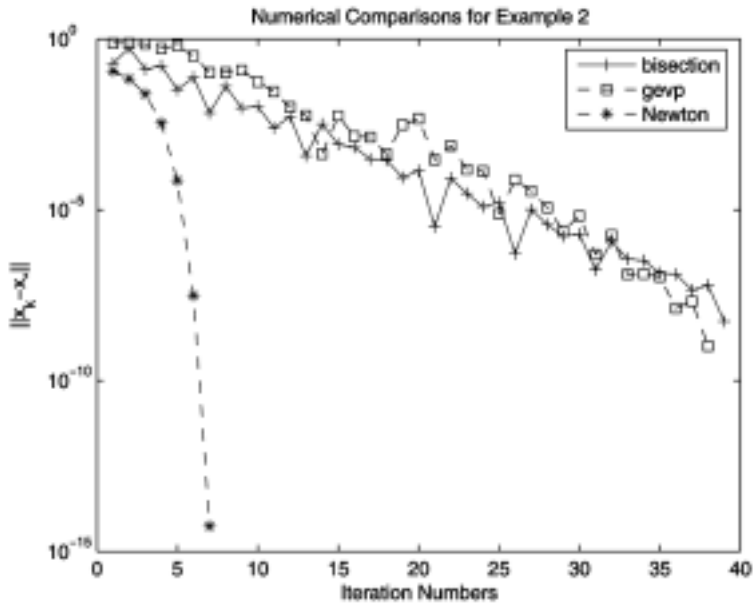

Fig. 8. Comparisons of the efficiency of numerical methods.

Example 3. The example is from[30]. Consider

$$M = \begin{bmatrix} 1+i & 10-2i & -20i \\ 5i & 3+i & -1+3i \\ -2 & i & 4-i \end{bmatrix}.$$

Since there is no exact solution to Example 3, we exploit the computation results by using the bisection method to obtain an acceptable solution,

$$\tilde{x} = \begin{bmatrix} 0.1565412174129063 \\ 0.4692003612185643 \\ 1.0000000000000000 \end{bmatrix}.$$

Since Newton's method is locally convergent, obtaining an appropriate initial guess is a crucial step for Newton's method. In this example, we adopt the iterative vector $x^{(k)}$ generated by the bisection method so that $\| \widehat{x}^{(k+1)} - \widehat{x}^{(k)} \| < 0.05$ or $\| \widehat{\nabla \lambda_{\max}}(x^{(k)}) \| < 0.05$, to be an initial vector for Newton's method.

Figs. 9, 10, illustrate the numerical results for Example 3 by applying the bisection and "bisection + Newton's" method, respectively, with initial guess $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$. Here "bisection + Newton's" method means that we use the bisection method first to obtain a reliable initial vector and then apply the Newton's method to compute the solution of the problem. Fig. 11 illustrate the numerical results for Example 3 by applying Newton's method with a good initial guess $\begin{bmatrix} 0.1 & 0.4 & 1.0 \end{bmatrix}^T$ which is closer to the target vector $\tilde{x}$. The results in Fig. 11 show the quadratic convergent behavior of Newton's method. Finally, we show the comparison of numerical results of proposed algorithms in Fig. 12. The results of comparison conclude that the method associated with `gevp` in MatLab LMI Toolbox which realize the interior point method for Problem 1 is very competitive with the bisection method. However, Newton's method is the most efficient algorithm for Example 3.

7. Conclusion

In this paper, we investigate the computation for the largest structured singular value, $\mu_{\mathbf{\Delta}}(M)$, under a diagonal uncertainty which can be seen to be an upper bound of $\mu_{\mathbf{\Delta}}(M)$ for the case of a general structured uncertainty. We develop a Newton's type method for computing the upper bound of $\mu_{\mathbf{\Delta}}(M)$. The related theoretical results are studied as well. Numerical implementation shows the efficiency for the developed Newton's method. In addition, the numerical results illustrate that the Newton's method converges local quadratically. Recently, the computation of the largest structured singular value $\mu_{\mathbf{\Delta}}(M)$ with various structured uncertainties, becomes an important problem [18, 19, 20, 23] because that the computation is the kernel issue for the optimal μ-synthesis controller designment which is more close

to the real-world application than $\mathcal{H}_2$ or $\mathcal{H}_\infty$ control. However, a fast algorithm for the computation of the largest structured singular value with various structured uncertainty is under investigated.

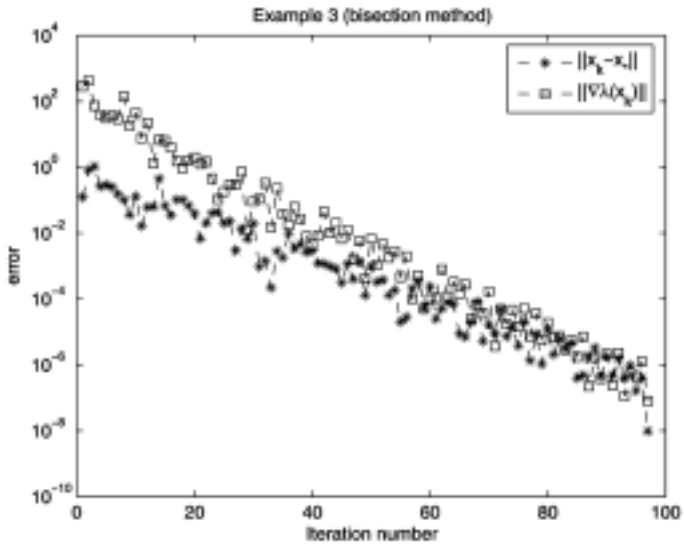

Fig. 9. Numerical results for Example 3 by using the bisection method.

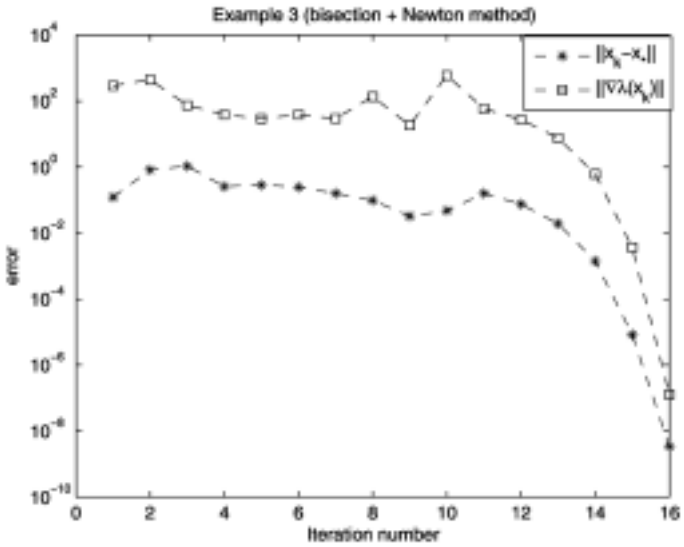

Fig. 10. Numerical results for Example 3 by using bisection + Newton's method.

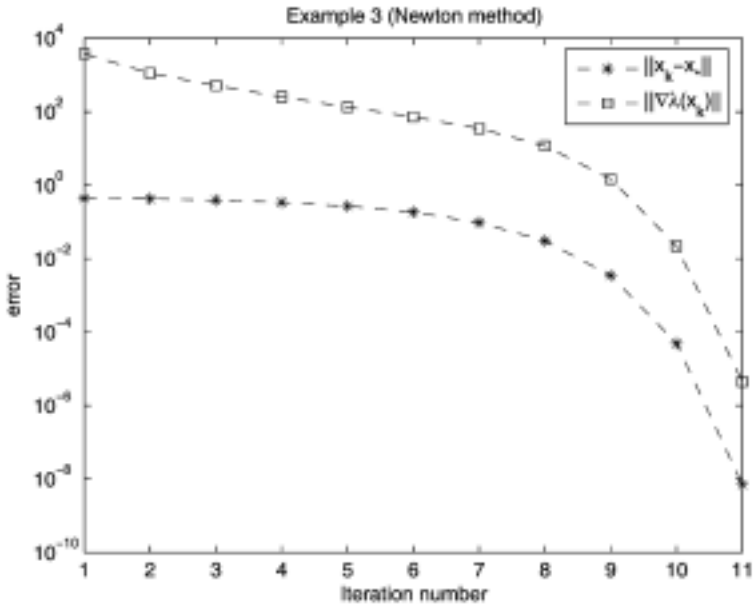

Fig. 11. Numerical results for Example 3 by Newton's method with initial guess $[0.1\ 0.4\ 1.0]^T$.

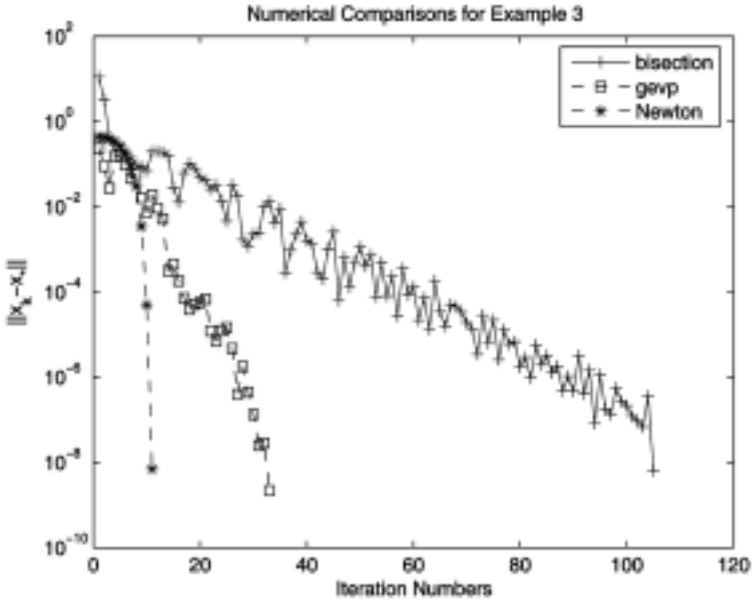

Fig. 12. Comparisons of the efficiency of numerical methods.

ACKNOWLEDGMENT

The authors are grateful to Professor Wen-Wei Lin of National Chiao Tung University, Taiwan, and Dr. Eric K. Chu of Monash University, Australia, for many helpful discussions and useful suggestions.

REFERENCES

1. A. L. Andrew, K. W. E. Chu and P. Lancaster, Derivatives of eigenvalues and eigenvectors of matrix function, *SIAM J. Matrix Anal. Appl.*, **14(4)** (1993), 903-926.
2. G. J. Balas, J. C. Doyle, K. Glover, A. Packard and R. Smith, *μ-Analysis and Synthesis Toolbox*, Math Works, Natick, MA, 1995, Chap. 4.
3. S. P. Boyd and V. Balakrishnan, A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm, *Systems and Control Letters*, **15** (1990), 1-7.
4. S. P. Boyd, L. Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, 1994.
5. R. Braatz, P. Young, J. Doyle and M. Morari, Computational complexity of μ calculation, *IEEE Transactions on Automatic Control*, **AC-39(5)** (1994), 1000-1002.
6. J. Bräuninger, A quasi-Newton method with Cholesky factorization, *Computing*, **25** (1980), 155-162.
7. R. H. Chan, S. F. Xu and H. M. Zhou, On the convergence rate of a quasi-newton method for inverse eigenvalue problems, *SIAM J. Numer. Anal.*, **36(2)** (1999), 436-441.
8. W. Cheney and D. Kincaid, *Numerical Mathematics and Computing*, Brooks/Cole, 1994.
9. J. C. Doyle, Analysis of feedback system with structured uncertainties, *IEEE Proc., Part D*, **129(6)** (1982), 242-250.
10. M. Fan, A. Tits and J. Doyle, Robustness in the presence of mixed parametric uncertainty and unmodelled dynamics, *IEEE Transactions on Automatic Control*, **AC-36(1)** (1991), 25-38.
11. S. C. Fang and S. Puthenpura, *Linear Optimization and Extensions*, Prentice Hall, N. J., 1993.
12. J. A. Ford and I. A. Moghrabi, Alternative parameter choices for multi-step quasi-newton methods, *Optim. Meth. Software*, **2** (1993), 357-370.
13. J. A. Ford and I. A. Moghrabi, Multi-step quasi-newton methods for optimization, *J. Comput. Appl. Math.*, **50** (1994), 305-323.
14. J. A. Ford and I. A. Moghrabi, Minimum curvature multi-step quasi-newton methods, *Computers Math. Applic.*, **31** (1996), 179-186.

15. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins U.P., 1996.

16. M. J. Hays, D. G. Bates and I. Postlethwaite, New tools for computing tight bounds on the real structured singular value, *Journal of Guidance, Control, and Dynamics*, **24(6)** (2001), 1204-1213.

17. H. N. Huang, M. L. Jiang, F. B. Yeh and L. C. Fu, *μ-synthesis via spectral interpolation theory*, Proc. of 2003 Chinese Automatic Control Conference and Bio-Mechatronics System Control and Application Workshop, Chung-Li, Taiwan, 2003, pp. 1809-1814.

18. C. T. Lawrence, A. L. Tits and P. Van Dooren, A fast algorithm for the computation of an upper bound on the μ-norm, *Automatica*, **36** (2000), 449-456.

19. J. Lee and T. F. Edgar, *Upper bounds of structured singular values for mixed uncertainties*, Proc. of the 42nd IEEE Conference on Decision and Control, Maui, Hawaii USA, 2003, pp. 798-802.

20. J. F. Magni and C. Döll, *μ-analysis for flexible systems*, ONERA-CERT, Département d'Automatique B. P. 4025, F31055 Toulouse Cedex 04.

21. MathWorks, *MATLAB user's guide*, The Math Works, Inc., 1992.

22. A. Megretski, *On the gap between structured singular values and their upper bounds*, Proc. of the 32nd Conference on Decision and Control, San Antonio, Texas, 1993, pp. 3461-3462.

23. M. L. Overton and P. Van Dooren, *On computing the complex passivity radius.* Proc. of the 44th IEEE Conference on Decision and Control and European Control Conference, Seville, Spain, 2005, pp. 7960-7964.

24. A. Packard and J. Doyle, The complex structured singular value, *Automatica*, **29(1)** (1993), 72-109.

25. A. Packard and J. Doyle, *A Power method for the structured singular value*, Proc. of the American Control Conference, **2**, Inst. of Electrical and Electronics Engineers, New York, 1988, pp. 1213-1218.

26. M. Safonov and P. Lee, *A multiplier method for computing real multivariable stability margins*, Proc. of the IFAC Word Congress, **1**, International Federation of Automatic Control, Oxford, 1993, pp. 275-278.

27. T. F. Sturm, Quasi-Newton method by Hermit interpolation, *J. Opti. Theory Appl.*, **83(3)** (1994), 587-612.

28. O. Toker and H. Özbay, On the complexity of purely complex μ computation and relate problems in multidimensional systems, *IEEE Trans. Auto. Control*, **43(3)**, (1998), 409-414.

29. P. M. Young, M. P. Newlin and J. C. Doyle, Computing bounds for the mixed μ problem, *International Journal of Robust and Nolinear Control*, **5(6)** (1995), 573-590.

30. K. Zhou, J. C. Doyle and K. Glover, *Robust and Optimal Control*, Prentice Hall, N.J., 1995.

Kun-Chu Chen
Department of Information Management,
National Kaohsiung University of Applied Science,
Kaohsiung, 807, Taiwan
E-mail: chenkj@cc.kuas.edu.tw

Chern-Shuh Wang and Ching-Chang Yen
Department of Mathematics,
National Cheng Kung University,
Tainan 701, Taiwan
E-mail: cswang@math.ncku.edu.tw
l1894104@mail.ncku.edu.tw